Identification of Natural Languages in the Limit: Exploring Frontiers of Finite Elasticity for General Combinatory Grammars

Erwan Moreau

FG 2007:
The 12th conference on Formal Grammar
Dublin, Ireland
August 4-5, 2007

Organizing Committee:

Laura Kallmeyer Paola Monachesi
Gerald Penn Giorgio Satta

CENTER FOR THE STUDY
OF LANGUAGE
AND INFORMATION

1

Identification of Natural Languages in the Limit: Exploring Frontiers of Finite Elasticity for General Combinatory Grammars

ERWAN MOREAU

Abstract

Kanazawa has proposed in Kanazawa (1998) several learnability results in Gold's model for categorial grammars. In this paper we explore the interests and limits of the method he used: we extend his main result to some subset of General Combinatory Grammars, and provide also a counter-example in order to locate the frontier of such a method.

Keywords GOLD'S MODEL, LEARNING CATEGORIAL GRAMMARS, FINITE ELASTICITY, GENERAL COMBINATORY GRAMMARS

1.1 Introduction

Children learn quite easily their mother tongue: they need only several years to master nearly all syntactic features of the language. But acquiring syntax of natural languages is a very hard task for the machine. More precisely, this problem consists in discovering rules that govern how sentences are built in a language. Gold's model of identification in the limit (Gold (1967)) is one of the main formalizations of the learning process: In this model, the learner has to guess the language given a set of examples (sentences). Input sentences may contain "structural information" or not, but the learner does never have "negative evidence" (i.e. sentences that do not belong to the language). This point is both the main advantage and the main drawback of the model:

FG-2007.
Organizing Committee:, Laura Kallmeyer, Paola Monachesi, Gerald Penn, Giorgio Satta.

indeed, this constraint is useful since obtaining an exhaustive set of wrong sentences for a natural language seems impossible. But of course it makes the learning process a lot harder, since in this case there is no counter-example in the sequence that would permit to correct an error.

In 1998, Kanazawa has proposed several promising results about learnability of categorial grammars in Gold's model Kanazawa (1998): in particular, he proved that k-valued AB grammars[1] (the simplest case of categorial grammars) are learnable from strings. His work was innovative in several respects: the use of the property of finite elasticity, different new techniques used to prove learnability, and an efficient algorithmic method for learning (only in some restricted cases). But above all, his learnability results were the first in Gold's model that concern a grammatical formalism which is quite well suited for the representation of natural languages. Indeed, most previous results in Gold's model were very theoretical ones about abstract classes of languages. More precisely, AB grammars are not totally suited to deal with natural languages, but belong to the family of categorial grammars, which is intended to represent natural languages in a reliable way.

For all these reasons, Kanazawa's work has given rise to various studies, extensions or variations of his results. The common point in all these studies is the use of Gold's model to learn (automatically) natural languages, generally using one of the various formalisms related to categorial grammars. Roughly, the idea was to see if it is possible to apply the "good learnability properties" of AB grammars to more general categorial grammars formalisms. However, results were not always positive, far from it. Actually, almost all learnability studies for richer grammatical formalisms came up against strong limitations.

One of the hypotheses would be that the good learnability results obtained by Kanazawa could be related to the logical properties of AB grammars. But the main results obtained with AB grammars do not apply to Lambek grammars, or only to some small subsets of these languages (Foret and Le Nir (2002)): this tends to show that learnability properties are not related to the "logical respects" of categorial grammars, and to discard most type logical grammatical formalisms from being learnable (at least in a general way).

Another hypothesis consists in considering AB grammars as a grammatical system based on a fixed set of rewriting rules using substitution. In this viewpoint, it is interesting to try to find what properties of AB grammars rules make them learnable, in order to see whether one may

[1]A grammar is k-valued if there are at most k distinct types assigned to each word in the lexicon.

obtain learnable classes by adding some other rules. Kanazawa has proposed a first result in this perspective, but only under some restricted conditions. His goal was mainly to study the case of Combinatory Categorial Grammars (Steedman (2000)), which is typically a categorial grammars system based on a fixed set of rewriting rules. He proposed a system called *General Combinatory Grammars* that allows to define any set of such universal rules for lexicalized grammars, and proved that rigid[2] structure languages are learnable for any set of such rules.

Actually, the main problem faced with this kind of learning method is the difference between the case of rigid structure languages and k-valued string languages: in the first case, learning is often possible (and even efficiently), but the rigidity constraint severely restricts the size of the class of languages, and the need for structures as input is a strong constraint (since this kind of structures is hardly available). On the contrary, k-valued string languages are very interesting classes, in particular for natural languages, but are not always learnable. For AB grammars, Kanazawa has shown that the property of finite elasticity, which is a sufficient condition for learnability, holds in both cases. More precisely, he proved that it is possible to extend finite elasticity from rigid structure languages to k-valued string languages. But finite elasticity does not hold for all classes of (rigid) GCG.

Thus, the problem of learning natural languages in Gold's model (using Kanazawa's method) could be summarized in the following way: one needs a richer grammatical formalism than AB grammars, in order to represent natural languages in a satisfying way (in a linguistic viewpoint). But it is also necessary not to be restricted to rigid structure languages, because their expressivity is too poor and such structures are too complex to be easily obtained as input. That is why it is interesting to see whether finite elasticity holds for some richer formalisms, because then it would be possible to use Kanazawa's method to show that the corresponding k-valued string languages are learnable. Since AB grammars is the only known such case among categorial grammars, it seems reasonable to try to find what gives this property to AB grammars rules: in the framework of General Combinatory Grammars, this may permit to discover new kind of rules satisfying these conditions, and thus potentially obtain new interesting learnable classes. This is the problem we address in this paper (subsection 1.3.2). We will also provide a negative result (subsection 1.3.3), to try to locate the frontier where such a method stops working.

[2]A grammar is rigid if there is only one type assigned to each word in the lexicon.

1.2 Learning Natural Languages in Gold's Model

1.2.1 Identification in the Limit

Gold's model of identification in the limit is a formal model of learning, introduced in Gold (1967). In this model, the learner has to guess the right language from an infinite sequence of objects belonging to this language. After each new example seen, the learner has to make an hypothesis about the language by proposing a grammar. Formally, let ϕ be a learning function, and L a language over a universe $\mathcal{U}$ ($L \subseteq \mathcal{U}$). Let $\langle a_i \rangle_{i \geq 0}$ be any infinite sequence of objects such that $a \in \langle a_i \rangle_{i \geq 0}$ if and only if $a \in L$. ϕ *converges* to a grammar G if there exists $n \geq 0$ such that for any $i \geq n$ $\phi(\langle a_1, a_2, \ldots, a_i \rangle) = G$. A class of languages $\mathcal{L}$ is learnable if there exists a learning function ϕ such that for all $L \in \mathcal{L}$, ϕ converges to a grammar G which corresponds to language L, for any enumeration $\langle a_i \rangle_{i \geq 0}$ of L.

Wright has proposed in Wright (1989) a sufficient condition for learnability, called *finite elasticity*. A class $\mathcal{L}$ has infinite elasticity if there exist two infinite sequences $a_0, a_1, \ldots$ of objects and $L_1, L_2, \ldots$ of languages such that for any $k \geq 1$ $\{a_0, a_1, \ldots, a_{k-1}\} \subseteq L_k$ but $a_k \notin L_k$. A class $\mathcal{L}$ has finite elasticity[3] if $\mathcal{L}$ does not have infinite elasticity. Kanazawa has shown an important theorem about finite elasticity in Kanazawa (1998): if a class $\mathcal{L} \subseteq \mathcal{U}$ has finite elasticity and there exists a *finite-valued*[4] relation R between universes $\mathcal{U}$ and $\mathcal{U}'$, then $\mathcal{L}' = \{ R^{-1}[L] \mid L \in \mathcal{L} \}$ has also finite elasticity.

Even if it is only a sufficient condition, finite elasticity is widely used to show learnability of classes of languages. Indeed, showing that there is no pair of infinite sequences of languages and objects verifying the infinite elasticity property is generally simpler than showing the existence of a converging learning algorithm for a complex class of languages.

1.2.2 AB Grammars and Finite elasticity

AB grammars, also called classical categorial grammars, are the most simple formalism among categorial grammars. An AB grammar G is a tuple $\langle \Sigma, Pr, \triangleright \rangle$ where Σ is the set of words and Pr the set of primitive types. The set of types Tp is defined as the smallest set such that $Pr \subseteq Tp$ and $A/B, A\backslash B \in Tp$ for all $A, B \in Tp$. The relation $\triangleright \subseteq \Sigma \times Tp$ assigns one or several types to each word. The relation $\rightarrow$ is defined with the following universal rules:

[3] Actually another condition is required for learnability: $\mathcal{L}$ must be an indexed family of recursive languages. This condition is easily fulfilled in the cases we study.

[4] A relation $R \subseteq \mathcal{U}_1 \times \mathcal{U}_2$ is *finite-valued* iff for every $a \in \mathcal{U}_1$ there are at most finitely many $b \in \mathcal{U}_2$ such that Rab (this property is not symmetric).

$$A/B,\ B \rightarrow A \text{ (for any } A, B \in Tp)$$
$$B,\ B\backslash A \rightarrow A \text{ (for any } A, B \in Tp)$$

A pair of types t, t' may be derived into a new type t'' using the FA rule, written $t\ t' \rightarrow_{FA} t''$, if there exists a substitution σ such that $\sigma(A/B) = t$, $\sigma(B) = t'$ and $\sigma(A) = t''$. Similarly, $t\ t' \rightarrow_{BA} t''$, if there exists σ such that $\sigma(B) = t$, $\sigma(B\backslash A) = t'$ and $\sigma(A) = t''$. $\rightarrow$ is defined as the union of $\rightarrow_{FA}$ and $\rightarrow_{BA}$.

The set of all *functor-argument structures* (FA-structures), denoted $SL(\mathcal{R}_{AB})$, is defined inductively in the following way: $\Sigma \subseteq SL(\mathcal{R}_{AB})$, and for any pair $T, T' \in SL(\mathcal{R}_{AB})$ the FA-structures $fa(T, T')$ and $ba(T, T')$ belong to $SL(\mathcal{R}_{AB})$. An FA-structure T is correct for a grammar G, denoted $T \in SL(G)$, if it is possible to label all nodes in T with a type $t \in Tp$, such that if t labels a leaf node $w \in \Sigma$, then $w \triangleright t$; otherwise, if t is a node $fa(T_1, T_2)$ (resp. $ba(T_1, T_2)$), then $t_1\ t_2 \rightarrow_{FA} t$ (resp. $t_1\ t_2 \rightarrow_{BA} t$), where t_1 and t_2 are respectively the types labelling the root of T_1 and T_2. Moreover, if t labels the root of T, then $t = s$. $w_1 \ldots w_n \in L(G)$ if and only if there exists a structure $T \in SL(G)$ with $\langle w_1 \ldots w_n \rangle$ the sequence of words occuring at its leaves.

In Buszkowski and Penn (1989), the authors proposed an algorithm called RG (*Rigid Grammars*). RG learns rigid AB grammars from FA-structures. Kanazawa (1998) provided a proof of convergence for RG and several results about finite elasticity of AB grammars:

Theorem 1 (Kanazawa)
$\{\ SL(G) \mid G$ *is a rigid AB grammar* $\}$ *has finite elasticity.*

Using his theorem about extending finite elasticity to another class with a finite valued relation, Kanazawa also showed the following result:

Corollary 1 (Kanazawa) *For any* $k \geq 0$*, let* $\mathcal{G}_k$ *be the set of all* k*-valued AB grammars.* $\{\ L(G) \mid G \in \mathcal{G}_k\ \}$ *has finite elasticity.*

As a consequence, k-valued AB grammars string languages are learnable. This result is important, because it is not restricted to rigid grammars and only simple strings are needed as input. However Costa Florêncio (2003) shows that both problems of learning k-valued AB grammars and of learning AB grammars from strings are NP-hard.

1.2.3 General Combinatory Grammars (GCG)

The name “General Combinatory Grammars” is used by Kanazawa to define any class of grammars using some set of operators and universal rules (expressed as the rewriting of a sequence of terms containing variables into another term). It refers to *combinatory categorial grammars* (CCG), defined by Steedman (2000), who proposed to add several rules to AB grammars, in order to obtain a better syntactic description of natural languages. This means that AB grammars, as well as CCG, are instances of GCG. It is worth noticing that GCG are "rule-based", and not related to the deductive "type-logical" approach of CG.

Given a set $\mathcal{S}$ of operators and a set $\mathcal{V}$ of variables, the set of $\mathcal{S}$-terms over $\mathcal{V}$ is the smallest set such that any $v \in \mathcal{V}$ is an $\mathcal{S}$-term over $\mathcal{V}$, and for any operator $f \in \mathcal{S}$ with $arity(f) = n$, if $t_1, \ldots, t_n$ are $\mathcal{S}$-terms over $\mathcal{V}$ then $f(t_1, \ldots, t_n)$ is an $\mathcal{S}$-term over $\mathcal{V}$.[5] Given a set $\mathcal{S}$ of operators and a set $Var(R)$ of variables, a universal rule R over $\mathcal{S}$ is any expression of the form $A_1, \ldots, A_n \rightarrow A_0$, where each A_i is an $\mathcal{S}$-term over $Var(R)$.

Definition 1 ($\mathcal{R}$-grammar) *Let $\mathcal{S}$ be a set of operators and $\mathcal{R}$ a set of universal rules over $\mathcal{S}$. An $\mathcal{R}$-grammar G is a system $\langle \Sigma, Pr, s, \triangleright \rangle$:*

- *Σ is the vocabulary,*
- *Pr is a finite set of variables, called primitive types. The set of types Tp is then defined as the set of $\mathcal{S}$-terms over Pr.*
- *s is an $\mathcal{S}$-term over $\emptyset$: this is the special type for correct sentences.*
- *$\triangleright$ is the relation assigning types to words: $\triangleright \subseteq \Sigma \times Tp$. Each pair $w \triangleright t$ in this relation is called a lexical rule.*

$|G|$ is the number of lexical rules in G, and $Lex(G)$ is the set of types used in the lexicon: $Lex(G) = \{ t \in Tp \mid \exists w \text{ such that } w \triangleright t \}$. $st(G)$ is the set of all subtypes in G: $st(G) = \{ t \mid \exists t' \in Lex(G) : t \text{ is a subtype of } t' \}$. $Var(G)$ is the set of all primitive types occuring at least once in G: $Var(G) = Pr \cap st(G)$. The set $SL(\mathcal{R})$ of $\mathcal{R}$-structures is defined inductively as: $\Sigma \subseteq SL(\mathcal{R})$, and for any rule $R = A_1 \ldots A_n \rightarrow A_0 \in \mathcal{R}$ and any sequence $S_1 \ldots S_n \in SL(\mathcal{R})$, $R(S_1, \ldots, S_n) \in SL(\mathcal{R})$.

Clearly an $\mathcal{R}$-structure is simply a generalization to GCG of the notion of FA-structure for AB grammars. In the same way, the relation $\rightarrow_R$ is defined for any rule $R = A_1 \ldots A_n \rightarrow A_0 \in \mathcal{R}$ as $t_1 \ldots t_n \rightarrow_R t_0$ if there exists σ such that $\sigma(A_i) = t_i$ for all i $(0 \leq i \leq n)$. An $\mathcal{R}$-structure belongs to $SL(G)$ if there exists a labelling of T which is correct w.r.t G: each leaf $w \in \Sigma$ labelled t verifies $w \triangleright t$, any node R labelled t with daughters $t_1 \ldots t_n$ verifies $t_1 \ldots t_n \rightarrow_R t$, and $root(T)$ is labelled s. A [complete] parse tree is an $\mathcal{R}$-structure "correctly labelled" with types [whose root is labelled s]. $ft(G)$ is the set of all types that may appear in a complete parse tree for G. G and G' are *equivalent*, denoted $G \equiv G'$, if G and G' are identical modulo a renaming of variables[6].

[5] The special case where $arity(f) = 0$ is included in this definition: if f is such an operator, then it is an $\mathcal{S}$-term over $\mathcal{V}$.

[6] In other words, there exist two substitutions σ and σ' such that $\sigma[G] = G'$ and $\sigma'[G'] = G$. Remark: $G \equiv G'$ implies $SL(G) = SL(G')$.

Kanazawa (1998) extends the RG learning algorithm to GCG:

Proposition 1 (Kanazawa)
If $D \subseteq D'$ then $RG(D) \sqsubseteq RG(D')$.
If $D \subseteq L$ then $SL(RG(D)) \subseteq L$.

Kanazawa showed that rigid $\mathcal{R}$-grammars $\mathcal{R}$-structure languages are learnable, for any set of rules $\mathcal{R}$. But he left open the question of finite elasticity for GCG.

1.3 Strengths and Weaknesses of the "AB Grammars Learning Approach"

1.3.1 Definitions

Definition 2 (Useful type) *Let G be an $\mathcal{R}$-grammar. A type $t \in Tp$ is useful for G if t is a subtype of s*[7] *or if there exists a rule $R \in \mathcal{R}$, $R = A_1 \dots A_n \rightarrow A_0$, and a substitution $\sigma : Var(R) \mapsto Tp$ such that*

- *$\sigma(A_i) \in ft(G)$ for any $i \geq 0$*
- *there exist an integer j ($0 \leq j \leq n$) and a subtype u in A_j such that $\sigma(u) = t$, and $u \in Var(R)$ if and only if t is primitive.*[8]

A type t is useless for G if t is not useful for G. A grammar G is said without useless types if any type $t \in st(G)$ is useful for G.

This definition may be roughly expressed in the following way: a type t is useful only if there exists a derivation in which it appears at a position where it "plays some role". Indeed, the existence of a subtype u in A_j such that $\sigma(u) = t$ guarantees that t may be unified at least with some part of type A_j.

Definition 3 ($\sqsubseteq$) *Let G and G' be two rigid grammars. $G \sqsubseteq G'$ if there exists a substitution σ such that $\sigma[G] \subseteq G'$. Moreover, $G \sqsubset G'$ if $G \not\equiv G'$.*

Let $arg_f : 1..arity(f) \mapsto \{0, 1\}$ be a function which defines argument positions for operator f. We consider only the case of *oriented* operators: an operator is oriented if there exists only one position $0 \leq i \leq n$ such that $arg_f(i) = 0$ (where n is the arity of f). We define below a set of useful functions for such operators.

Definition 4 *Let $\mathcal{S}$ be a set of oriented operators and t an $\mathcal{S}$-term:*

- *if $t \in Pr$, then $pr(t) = t$. Otherwise, let $t = f(t_1, \dots, t_n)$ and let h be the unique position such that $arg_f(h) = 0$: $pr(t) = t_h$.*
- *$head(t) = t$ if $t \in Pr$, otherwise $head(t) = head(pr(t))$.*

[7] Remark: in this case t does not contain any type from Pr.
[8] Remark: this condition is equivalent to "u is primitive implies t is primitive".

- *$head^*(t) = \{t\}$ if $t \in Pr$, else $head^*(t) = \{t\} \cup head^*(pr(t))$.*
- *$args(t) = \emptyset$ if $t \in Pr$, otherwise let $t = f(t_1, \ldots, t_n)$: $args(t) = \{ t_i \mid arg_f(i) = 1 \}$.*
- *$args^*(t) = \emptyset$ if $t \in Pr$, else $args^*(t) = args(t) \cup args^*(pr(t))$.*

Definition 5 ((Strictly) arguments consuming rules) *Let $\mathcal{S}$ be a set of oriented operators and $R = A_1 \ldots A_n \to B$ a rule over $\mathcal{S}$.*

A_h is a consumer type for the rule R if:

- *$B \in head^*(A_h)$;*
- *$args^*(A_h) \subseteq Var(R)$;*
- *For any $i \neq h$, $A_i \in args^*(A_h)$;*
- *If $v \in args^*(A_h)$ such that $A_i \neq v$ for all i then there is only one occurrence of v in A_h.*

$R = A_1 \ldots A_n \to B$ is an arguments consuming rule if there exists an integer h, $1 \leq h \leq n$, such that A_h is a consumer type for R.

Moreover, R is a strictly arguments consuming rule if $B \neq A_h$.

The consumer type, if it exists, is unique. $Heads(G)$ is defined as:

$$Heads(G) = \{ t \in Pr \mid \exists t' \in Lex(G) : t = head(t') \}.$$

Example 1

$$\mathcal{R}_? = \left\{ \begin{array}{lll} (R_{/^?_1}) & A/^?B \quad B \to A & (with\ Var(R_{/^?_1}) = \{A, B\})\ ; \\ (R_{/^?_\varepsilon}) & A/^?B \to A & (with\ Var(R_{/^?_\varepsilon}) = \{A, B\}) \end{array} \right\}$$

With usual definitions $arg_{/^?}(1) = 0$, $arg_{/^?}(2) = 1$, these rules also verify all conditions of strictly arguments consuming rules. But rules like $\{ A/^*B\ B \to A/^*B\ ;\ A/^*B \to A \}$ are only arguments consuming rules (not strictly *ones).*

Proposition 2 *Let $G = \langle \Sigma, Pr, s, \triangleright \rangle$ be an $\mathcal{R}$-grammar, with $\mathcal{R}$ a set of arguments consuming rules. If there exists a (partial) parse tree P such that $root(P) = t$, then there exists $t' \in Lex(G)$ and a leaf labelled w in P such that $w \triangleright_G t'$ and $t \in head^*(t')$.*

1.3.2 Extending Finite Elasticity to GCG

In this section we propose a sufficient condition for finite elasticity of $\mathcal{R}$-grammars. This condition is given as a set of restrictions over universal rules[9], which makes it quite easy to verify for any class of GCG. Due to lack of space, below we provide only the main steps to show theorem 1 (the detailed proof may be found in Moreau (2006)).

[9] Costa Florêncio has given another sufficient condition in Costa Florêncio (2003), but his result is very different because the condition is expressed in an intermediate formalism. Since the required transformation does not preserve any limit on the number of rules, the interest of his result for k-valued grammars is quite limited.

Lemma 1 *Let $\mathcal{R}$ be a set of arguments consuming rules, G an $\mathcal{R}$-grammar without useless types and σ a substitution such that $\sigma[G]$ does not contain any useless type and $|Heads(G)| = |Heads(\sigma[G])|$. The three following propositions hold:*

1. *For any $x, x' \in Heads(G)$, $x \neq x' \Rightarrow head(\sigma(x)) \neq head(\sigma(x'))$.*
2. *Let $x \in Heads(G)$ and $t \in Lex(\sigma[G])$: if there exists $x' \in head^*(\sigma(x))$ such that $x' \in head^*(t)$, then $\sigma(x) \in head^*(t)$.*
3. *For any $x \in Heads(G)$, if $x' \in head^*(pr(\sigma(x)))$ then $x' \neq s$ and $x' \neq \sigma(t)$ for any $t \in st(G)$.*

Lemma 2 *Let $\mathcal{R}$ be a set of strictly arguments consuming rule. If G is an $\mathcal{R}$-grammar without useless types, σ a substitution such that $\sigma[G]$ does not contain any useless type and $|Heads(G)| = |Heads(\sigma[G])|$,*

then $|head^*(\sigma(x))| \leq M_{\mathcal{R}}$, for any $x \in Heads(G)$,

with $M_{\mathcal{R}} = \max(\{\, |head^*(A_0)| \mid \exists\, R = A_1 \ldots A_n \to A_0 \in \mathcal{R}\,\})$.

Proof.(sketch) Under these hypotheses, one can show using lemma 1 that for any (partial) parse tree P for $\sigma[G]$, if there exists $x \in Heads(G)$ such that there exists a node labelled x' in P, with $x' \in head^*(pr(\sigma(x)))$, then there exists $y \in Heads(G)$ such that $root(P) \in head^*(pr(\sigma(y)))$. Then it may be shown that for any $x \in Heads(G)$, if there exists $x' \in head^*(pr(\sigma(x)))$, then $x' \notin ft(\sigma[G])$. Therefore one obtains a bound on the size of $\sigma(x)$. □

Proposition 3 *Let $\mathcal{R}$ be a set of strictly arguments consuming rules. There exists no infinite sequence $G_1 \sqsubset G_2 \sqsubset \ldots$ of rigid $\mathcal{R}$-grammars without useless types over Σ such that for any $i > 0$:*

- *there exists σ_i such that $\sigma_i[G_i] = G_{i+1}$,*
- *$|Heads(G_i)| = |Heads(G_{i+1})|$,*
- *and $head(\sigma_i(x)) = \sigma_i(x)$ for any $x \in Heads(G_i)$.*

Proposition 4 *Let $\mathcal{R}$ be a set of strictly arguments consuming rules. There exists no infinite sequence $G_1 \sqsubset G_2 \sqsubset \ldots$ of rigid $\mathcal{R}$-grammars without useless types over Σ such that for any $i > 0$ there exists σ_i such that $\sigma_i[G_i] = G_{i+1}$, and $|Heads(G_i)| = |Heads(G_{i+1})|$.*

Proof. Suppose such an infinite sequence exists.

$$\text{Let } p_i = \sum_{x \in Heads(G_i)} |head^*(\sigma_i(x))|.$$

From lemma 2, there exists a constant $M_{\mathcal{R}}$ such that $|head^*(\sigma_i(x))| \leq M_{\mathcal{R}}$ for any $i > 0$ and any $x \in Heads(G_i)$, so $p_i \leq M_{\mathcal{R}} \times |G_i|$. It is clear that $|head^*(x)| \leq |head^*(\sigma_i(x))|$ for any $x \in Heads(G_i)$, therefore we must have $p_i \leq p_{i+1}$. Then there must exists $j_0 > 0$ such that $p_i = p_{j_0}$

for any $i > j_0$. This is possible only if $|head^*(x)| = |head^*(\sigma_i(x))|$, in other words if $\sigma_i(x) = head(\sigma_i(x))$ for all $x \in Heads(G_i)$ and any $i \geq j_0$. Thus the existence of the sequence $G_{j_0} \sqsubset G_{j_0+1} \sqsubset \ldots$ contradicts proposition 3. □

Proposition 5 *Let $\mathcal{R}$ be a set of strictly arguments consuming rules. There exists no infinite sequence $G_1 \sqsubset G_2 \sqsubset \ldots$ of rigid $\mathcal{R}$-grammars without useless types such that $|G_i| = |G_j|$ for any $i, j > 0$.*

Proof. (sketch) $G_i \sqsubset G_{i+1}$ implies that there exists σ_i such that $\sigma_i[G_i] \subseteq G_{i+1}$. Moreover, $\sigma_i[G_i] = G_{i+1}$ because $|G_i| = |G_{i+1}|$. One can show that $i \leq j$ implies $|Heads(G_i)| \geq |Heads(G_j)|$, so there must be an infinite subsequence such that $|Heads(G_i)|$ is constant. This contradicts proposition 4. □

Corollary 2 *Let $\mathcal{R}$ be a set of strictly arguments consuming rules. There exists no infinite sequence $G_1 \sqsubset G_2 \sqsubset \ldots$ of rigid $\mathcal{R}$-grammars without useless types over Σ.*

Theorem 1 *Let $\mathcal{R}$ be a set of strictly arguments consuming rules. Let $\mathcal{G}$ be any class of rigid $\mathcal{R}$-grammars over a vocabulary Σ.*
$SL(\mathcal{R})_{\mathcal{G}} = \{ SL(G) \mid G \in \mathcal{G} \}$ has finite elasticity.

Proof. Suppose $SL(\mathcal{R})_{\mathcal{G}}$ does not have finite elasticity: there exist an infinite sequence $T_0, T_1, T_2, \ldots$ of structures and an infinite sequence $L_1, L_2, \ldots$ of languages such that $\{T_0, \ldots, T_{i-1}\} \subseteq L_i$ and $T_i \notin L_i$ for any $i > 0$.

Let $G_i = RG(\{T_0, \ldots, T_{i-1}\})$ for any $i > 0$: from proposition 1, $G_i \sqsubseteq G_{i+1}$. By definition, G_i does not contain any useless type. Moreover, $SL(G_i) \subseteq L_i$ (prop. 1), so $T_i \notin SL(G_i)$. Therefore $G_i \not\equiv G_{i+1}$ for any $i > 0$, so there is an infinite sequence $G_1 \sqsubset G_2 \sqsubset \ldots$ of rigid grammars without useless types. This contradicts corollary 2. □

Using Kanazawa's method (finite-valued relation between both classes), this result is extended to k-valued $\mathcal{R}$-grammars string languages:

Corollary 3 *Let $\mathcal{R}$ be a set of strictly arguments consuming rules with at least two types in the left handside*[10] *and k be a positive integer: $\{ L(G) | G$ is a k-valued grammar$\}$ has finite elasticity and is learnable.*

By applying corollary 3, one may show that some new classes of k-valued $\mathcal{R}$-grammars are learnable from strings. But most "usable" strictly arguments consuming rules are very close to AB grammars ones (see example 1). It is possible to build more complex rules: for

[10] This constraint is required for a finite-valued relation to exist between string and structure languages.

example, a rule like $A\ B\ f(A, g(D, g(C, E)), B, D)\ D \to g(C, E)$ fulfils the conditions (with suitable definitions for arg_f and arg_g), but it seems hard to find rules fulfilling conditions *and* having a linguistic interest.

1.3.3 Where Finite Elasticity does not hold

The result we presented above is a sufficient condition, therefore its poorness is not significant alone. That is why we propose below a negative result to complete it.

Example 2 *Let $\mathcal{R}_0$ be the following set of rules:*

$$\begin{array}{ll} / & A/B\ B \to A \\ /_i^+ & A/^+B\ B \to A/^+B \\ /_\varepsilon^+ & A/^+B\ B \to A \end{array}$$

Let $G_0 = \{\ x : s/a_0\ ;\ y : s/b_0\ ;\ z : a_0\ ;\ w : b_0\ \}$, and for any $i > 0$ let σ_i be defined as:

$$\sigma_i = \begin{cases} \{\ a_{i-1} \mapsto a_i/^+b_i\ ;\ b_{i-1} \mapsto b_i\ \} \text{ if } i \text{ is odd ;} \\ \{\ a_{i-1} \mapsto a_i\ ;\ b_{i-1} \mapsto b_i/^+a_i\ \} \text{ if } i \text{ is even.} \end{cases}$$

For any $i > 0$, let $G_i = \sigma_i[G_{i-1}]$.

Lemma 3 *For any $i > 0$, there exists a complete parse tree for G_i containing a subtree with root a_i (resp. $a_i/^+b_i$) and a subtree with root $b_i/^+a_i$ (resp. b_i) if i is even (resp. odd).*

Using lemma 3, one may build an infinite sequence of $\mathcal{R}_0$-grammars $\langle G'_i \rangle_{i \geq 0}$ and an infinite sequence of $\mathcal{R}_0$-structures $\langle e_i \rangle_{i \geq 0}$ such that $\{e_0, \ldots, e_{i-1}\} \subseteq SL(G'_i)$ and $e_i \notin SL(G'_i)$ for any $i > 0$. Therefore:

Proposition 6 *$\{SL(G) | G$ is a rigid $\mathcal{R}_0$-grammar$\}$ has infinite elasticity.*

$\mathcal{R}_0$ is a set of arguments consuming rules. Therefore this proposition shows that there exist classes of arguments consuming grammars that do not have finite elasticity, whereas this property holds for all classes of *strictly* arguments consuming grammars. Thus one can see that, by simply removing one single constraint in the sufficient condition, finite elasticity does not hold anymore.

1.4 Discussion

The constraints of strictly arguments consuming rules are strong: there are only very few interesting grammatical formalisms that verify all these conditions. Actually, there may be larger GCG classes having also finite elasticity. But the counter-example we provide in proposition 6 permits to interpret in a more general way this result: it shows that crossing the narrow line between strictly arguments consuming grammars and non-strictly ones is enough to lose finite elasticity (in general). Since we have followed faithfully Kanazawa's method, this probably means that such a method can not be extended further: Kanazawa's good results depend on the particular case of AB grammars, or more precisely on some specificities of AB grammars that very few formalisms share. So even if Kanazawa's results are at first sight an important first step towards learnability of categorial grammars, it appears that they are very hard to extend to richer formalisms.

Of course, there may be a lot of methods to show learnability in Gold's model, so maybe one will find interesting learnable classes of categorial grammars in the future. In our viewpoint, Kanazawa's main contribution to learnability of natural languages in Gold's model lies in the generalized use of structural informations as input. This point is essential in his results and brought a new way to study learnability, especially for natural languages. But, in our opinion, his work should not be interpreted as an evidence that Gold's model is well suited to learn automatically natural languages: this question remains open.

References

Buszkowski, Wojciech and Gerald Penn. 1989. Categorial grammars determined from linguistic data by unification. Tech. Rep. TR-89-05, Department of Computer Science, University of Chicago.

Costa Florêncio, Christophe. 2003. *Learning categorial grammars*. Ph.D. thesis, Utrecht University.

Foret, Annie and Yannick Le Nir. 2002. On limit points for some variants of rigid lambek grammars. In *ICGI'2002, the 6th International Colloquium on Grammatical Inference*, no. 2484 in LNAI. Springer-Verlag.

Gold, E.M. 1967. Language identification in the limit. *Information and control* 10(5):447–474.

Kanazawa, Makoto. 1998. *Learnable classes of categorial grammars*. Cambridge University Press. ISBN 1-57586-097-X.

Moreau, Erwan. 2006. *Acquisition de grammaires lexicalisées pour les langues naturelles*. Ph.D. thesis, Université de Nantes.

Steedman, Mark. 2000. *The Syntactic Process*. Cambridge, Massachusetts: The MIT Press. ISBN 0-262-19420-1.

Wright, K. 1989. Identification of unions of languages drawn from an identifiable class. In *Proceedings of the Second Annual Workshop on Computational Learning Theory*, pages 328–333. Morgan Kaufmann.